

Title	デフレーションを前処理とするGMRES(\$m\$)法 (数値計算における前処理の研究)
Author(s)	森屋, 健太郎; 野寺, 隆
Citation	数理解析研究所講究録 (1999), 1084: 72-86
Issue Date	1999-02
URL	http://hdl.handle.net/2433/62784
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

デフレーションを前処理とする GMRES(m) 法

慶應義塾大学理工学部 森屋 健太郎 (Kentaro Moriya)

慶應義塾大学理工学部 野寺 隆 (Takashi Nodera)

概要

GMRES(m) 法は、非対称の大型で疎な連立 1 次方程式の非定常反復解法のひとつである。しかし、反復の途中でリスタートをすることによって、近似解を構成している固有ベクトルの情報が欠落してしまい、精度の良い近似解を得られなくなることがしばしばある。そこで、我々は固有ベクトルの情報を付加する 3 つの GMRES(m) 法について考察し、それらの算法の性能を比較評価する。富士通の分散メモリ型並列計算機 AP3000 による数値実験の結果から、3 つの算法のうち DEFLATED-GMRES(m, k) 法がもっとも良い性能を発揮することを示す。

1 はじめに

理工学のさまざまな諸現象のシミュレーションや解析は、連立 1 次方程式

$$Ax = b, \quad A \in R^{n \times n} \quad (1)$$

(ただし、 A は大型で疎な正則行列) を解くことに帰着されることが多い。従って、連立 1 次方程式 (1) の近似解を求めるための効率の良い反復解法が求められている。もし A が対称正定値であるなら、共役勾配法 (CG 法ともいう) [1] がもっとも効果的な反復解法であるということが知られている。しかし、 A が非対称の場合には、まだ決定的な反復解法はなく、現在さまざまな研究が行なわれている。

GMRES 法 [3] は、 A が非対称である場合の代表的な非定常反復解法のひとつであるが、 A の次元数と同じ数だけ正規直交ベクトルを生成する、いわゆる完全直交化を必要とするので、行列 A が大規模のときには非現実的な算法となる。それに対して、GMRES(m) 法は、クリロフ部分空間に生成する正規直交ベクトルの本数がある小さな数 m に制限することによって、正規直交化にかかる計算量と記憶領域のコストを減少させる算法である。GMRES(m) 法では、各 m 回の反復ごとにリスタートをして正規直交化を再び行なうが、このリスタートが原因で、近似解を構成している固有ベクトルの情報が欠落することがある [11]。この場合、残差ノルムは停滞現象を起こして精度の良い近似解が求まらなくなってしまう。

本稿では、 A の固有ベクトルの情報を付加することで残差ノルムの収束を向上させる 3 つの GMRES(m) 法 [7, 10, 11] について考察する。また、それらの算法を富士通分散メモリ型並列計算機 AP3000 に実装して、反復解法としての性能を GMRES(m) 法などと比較評価する。

```

GMRES( $m$ )

choose  $x_0$ 
 $r_0 = b - Ax_0$ 
 $\beta = \|r_0\|, \quad v_1 = r_0/\beta$ 
start:
for  $i = 1$  to  $m$  do
     $q = Av_i$ 
    for  $j = 1$  to  $i$  do
         $h_{j,i} = (v_j, q)$ 
    enddo
     $\bar{v}_{i+1} = q - \sum_{j=1}^i h_{j,i} v_j$ 
     $h_{i+1,i} = \|\bar{v}_{i+1}\|, \quad v_{i+1} = \bar{v}_{i+1}/h_{i+1,i}$ 
enddo
compute  $y_m = \min_y \|\beta e_1 - \bar{H}_m y\|$ 
 $x_m = x_0 + V_m y_m$ 
 $r_m = b - Ax_m$ 
if  $\|r_m\| / \|r_0\| < \varepsilon$  then
    stop iteration
endif
 $r_0 = r_m, \quad x_0 = x_m$ 
 $\beta = \|r_0\|, \quad v_1 = r_0/\beta$ 
goto start

```

図 1 GMRES(m) 法

2 GMRES(m) 法

ここでは GMRES(m) 法の実装法のみを示すので、算法の詳細は文献 [3] を参照してほしい。GMRES(m) 法は、アーノルディ過程から m 本の正規直交ベクトル

$$V_m = (v_1, v_2, \dots, v_m) \quad (2)$$

を生成する。近似解は、この m 本の正規直交ベクトルによって構成されるので、

$$x_m = x_0 + V_m y_m \quad (3)$$

となる。式 (3) の係数ベクトル y_m は、残差ノルムの最小 2 乗問題

$$y_m = \min_y \|\beta e_1 - \bar{H}_m y\| \quad (4)$$

$(\beta = \|r_0\|)$ を解くことで求められる。図 1 に GMRES(m) の算法を示すことにする。ここで、 \bar{H}_m はアーノルディ過程の副産物として生じる $(m+1) \times m$ の上ヘッセンベルグ行列であり、 ij 成分 ($j < i+2$) に $h_{j,i}$ を持つ。GMRES(m) 法は、GMRES 法と違い直交ベクトルの本数がある小さな値 m に制限されているので、計算機に実装可能な算法となっている。しかし、リスタートが原因で近似解を構成する固有ベクトルの情報が欠けてしまい、残差ノルムの収束が停滞することがしばしばある [11]。

```

MORGAN( $m, k$ )

choose  $x_0$ 
 $r_0 = b - Ax_0$ 
 $\beta = \|r_0\|, \quad v_1 = r_0/\beta$ 
 $U_k = \{\}$ 
start:
for  $i = 1$  to  $m$  do
   $q = Av_i$ 
  for  $j = 1$  to  $i$  do
     $h_{j,i} = (v_j, q)$ 
  enddo
   $\bar{v}_{i+1} = q - \sum_{j=1}^i h_{j,i} v_j$ 
   $h_{i+1,i} = \|\bar{v}_{i+1}\|, \quad v_{i+1} = \bar{v}_{i+1}/h_{i+1,i}$ 
enddo
for  $i = m+1$  to  $p$  do
   $q = Au_{i-m}$ 
  for  $j = 1$  to  $i$  do
     $h_{j,i} = (v_j, q)$ 
  enddo
   $\bar{v}_{i+1} = q - \sum_{j=1}^i h_{j,i} v_j$ 
   $h_{i+1,i} = \|\bar{v}_{i+1}\|, \quad v_{i+1} = \bar{v}_{i+1}/h_{i+1,i}$ 
enddo
compute  $y_p = \min_y \|\beta e_1 - \bar{H}_p y\|$ 
 $x_p = x_0 + W_p y_p$ 
 $r_p = b - Ax_p$ 
if  $\|r_p\| / \|r_0\| < \varepsilon$  then
  stop iteration
endif
 $r_0 = r_p, \quad x_0 = x_p$ 
 $\beta = \|r_0\|, \quad v_1 = r_0/\beta$ 
compute  $k$  eigenvectors  $U_k$  from (11)
goto start

```

図 2 MORGAN(m, k) 法

3 固有ベクトルの情報を付加した GMRES(m) 法

ここでは、固有ベクトルの情報を利用する 3 種類の GMRES(m) 法について考察する。これらの算法は大きく 2 つの部類に分けることができる。ひとつは、正規直交基底に固有ベクトルを追加して、その拡張部分空間から近似解を構成する手法である。もうひとつは、固有ベクトルによって前処理行列を構成する手法である。MORGAN(m, k) 法は前者に属し、DEFLATED-GMRES(m, k) 法及び DEFLATION(m, k) 法は後者に属する。

3.1 MORGAN(m, k) 法

A の k 本の固有ベクトル u_1, \dots, u_k を正規直交基底 V_m に付加して、拡張部分空間

$$W_p = (v_1, \dots, v_m, u_1, \dots, u_k) \quad (5)$$

を生成する [7, 11]. ただし, $p = m + k$ である. この拡張部分空間に対してアーノルディ過程を適用すると p 本の正規直交基底

$$V_p = (v_1, \dots, v_m, v_{m+1}, \dots, v_p) \quad (6)$$

が生成される. 近似解は拡張部分空間 W_p 上に構成されるので,

$$x_p = x_0 + W_p y_p \quad (7)$$

となる. y_p は式 (4) の最小 2 乗問題から求められるが, この場合, 最小 2 乗問題の次元は m でなく p となる. よって, アーノルディ過程から生成されるヘッセンベルグ行列も $(p+1) \times p$ となるので, \bar{H}_p と書き表わすことができる.

次に, A の固有ベクトルの求め方について示す. A の固有値問題は任意の p 次元のベクトルを用いて,

$$\begin{cases} AW_p g_p = \theta W_p g_p \\ u = W_p g_p \end{cases} \quad (8)$$

と書き表すことができる. ここで, u が A の固有ベクトルに相当する. 式 (8) の上式の両辺に W_p^* を掛けることで, 式 (8) を小さな次元 p の一般固有値問題に変換することができるが, $(AW_p)^*$ を両辺に掛けるほうが精度の良い固有ベクトルが求まることが知られている [5]. 従って, 式 (8) の上式の両辺に $(AW_p)^*$ を掛けると,

$$(AW_p)^* AW_p g_p = \theta (AW_p)^* W_p g_p \quad (9)$$

となる. さらに,

$$AW_p = V_{p+1} \bar{H}_p \quad (10)$$

の関係があるので, 式 (9) の左辺は

$$\begin{aligned} (AW_p)^* AW_p &= (V_{p+1} \bar{H}_p)^* V_{p+1} \bar{H}_p \\ &= \bar{H}_p^* V_{p+1}^* V_{p+1} \bar{H}_p \\ &= \bar{H}_p^* \bar{H}_p \end{aligned}$$

となる. 従って, 一般固有値問題

$$\begin{cases} \bar{H}_p^* \bar{H}_p g_p = \theta (AW_p)^* W_p g_p \\ u = W_p g_p \end{cases} \quad (11)$$

から A の固有ベクトルを求めることができる. また, 式 (11) における行列 $(AW_p)^* W_p$ の最初の $m \times m$ のブロックは, H_m^* (H_m は \bar{H}_m の最後の行を取り除いたもの) に等しい. 以上のことから, 式 (11) の一般固有値問題において, 左辺と右辺の行列を求める計算量を減少させることができる. 図 2 に MORGAN(m, k) 法の算法を示す. 固有ベクトルは, k 個の小さい固有値に対応するものを付加する. 従来の GMRES(m) 法との違いは, 行列 A と固有ベクトルの積 Au_i ($1 \leq i \leq k$) にアーノルディ過程を適用して正規直交基底を k 次元だけ拡張していることと, 各リスタート毎に式 (11) の一般固有値問題から k 本の固有ベクトルを求めていることである.

DEFLATED-GMRES(m, k)

```

choose  $x_0$ 
 $r_0 = b - Ax_0$ 
 $\beta = \|r_0\|, \quad v_1 = r_0/\beta$ 
 $U_l = \{\}$ 
 $l = 0$ 
 $M^{-1} = I_n$ 
start:
for  $i = 1$  to  $m$  do
   $q = AM^{-1}v_i$ 
  for  $j = 1$  to  $i$  do
     $h_{j,i} = (v_j, q)$ 
  enddo
   $\bar{v}_{i+1} = q - \sum_{j=1}^i h_{j,i} v_j$ 
   $h_{i+1,i} = \|\bar{v}_{i+1}\|, \quad v_{i+1} = \bar{v}_{i+1}/h_{i+1,i}$ 
enddo
compute  $y_m = \min_y \|\beta e_1 - \bar{H}_m y_m\|$ 
 $x_m = x_0 + M^{-1}V_m y_m$ 
 $r_m = b - AM^{-1}x_m$ 
if  $\|r_m\| / \|r_0\| < \varepsilon$  then
  stop iteration
endif
 $r_0 = r_m, \quad x_0 = x_m$ 
 $\beta = \|r_0\|, \quad v_1 = r_0/\beta$ 
if  $l < k$  then
  compute  $f$  eigenvectors  $U_f$  from (16)
   $U_l = \text{orthog}(U_f, U_l)$ 
   $l = l + f$ 
   $T_l = U_l^* A U_l$ 
   $M^{-1} = I_n + U_l(|\lambda_n|T_l - I_l)U_l^*$ 
endif
goto start

```

図 3 DEFLATED-GMRES(m, k) 法**3.2 DEFLATED-GMRES(m, k) 法**

A の固有ベクトルの基底

$$U_l = (u_1, u_2, \dots, u_l) \quad (12)$$

によって、前処理行列

$$M^{-1} = I_n + U_l \left(|\lambda_n| T_l^{-1} - I_l \right) U_l^* \quad (13)$$

を構成する [10, 11]. M^{-1} を A の右側から組み込み, $AM^{-1}v_i$ ($1 \leq i \leq m$) に対してアーノルディ過程を適用する. ただし, $T_l = U_l^* A U_l$ であり, U_l は正規直交化されているものとする. また, $|\lambda_n|$ は A の最大固有値に相当する. U_l にはリスタート毎に f (ただし, $f \leq k$ かつ $k \bmod f = 0$) 本の固有ベクトルが $l = k$ となるまで付加されて, M^{-1} が更新されていく. 固有ベクトルの計算は, 最初の数回のリスタートでのみ必要となる. M^{-1} が l 本の固有ベクトルによって構成されているとき, AM^{-1} は, l 個の小さい固有値の変わりに最大固有値 $|\lambda_n|$ を l 個重複して持つことに

なる。従って、 l が大きくなるほど M^{-1} の前処理としての性能は向上していくことになる。

次に、固有ベクトルの求め方について考える。 m 次元の任意のベクトルを g_m とすると、 AM^{-1} に関する固有値問題は

$$\begin{cases} AM^{-1}V_m g_m = \theta V_m g_m \\ u = V_m g_m \end{cases} \quad (14)$$

となる。 u が AM^{-1} の固有ベクトルである。式 (14) の上式の両辺に V_m^* を掛けることで、 V_m の正規直交性から、式 (14) の上式を

$$V_m^* AM^{-1} V_m g_m = \theta g_m \quad (15)$$

に変換することができる。さらに、 $V_m^* AM^{-1} V_m = H_m$ (H_m は \bar{H}_m の最後の行を取り除いたもの) の関係があるので、式 (15) の左辺は H_m に置き換えることができる。従って、 AM^{-1} の固有ベクトルは、標準固有値問題

$$\begin{cases} H_m g_m = \theta g_m \\ u = V_m g_m \end{cases} \quad (16)$$

を解くことで求めることができる。図 3 に DEFLATED-GMRES(m, k) 法の算法を示す。従来の GMRES(m) 法との違いは、アーノルディ過程に右側前処理行列 M^{-1} が組み込まれていることと、付加する固有ベクトルが k 本に達するまで、式 (16) の標準固有値問題から固有ベクトルを求めて、前処理行列 M^{-1} を更新することである。

3.3 DEFLATION(m, k) 法

DEFLATION(m, k) 法は、DEFLATED-GMRES(m, k) 法と同じ前処理行列を構成することになるが、前処理を構成する固有ベクトルの求め方が異なる [11]。まず、DEFLATED-GMRES(m, k) 法と同様に、リスタート毎に f (ただし、 $f \leq k$ かつ $k \bmod f = 0$) 本の固有ベクトルを基底 U_l が $k + f$ 次元になるまで付加していく。しかし、これらの固有ベクトルを求めるのに、DEFLATED-GMRES(m, k) 法と同じ手順は踏まず、次のような手順に従う。まず、 A に関する固有値問題

$$\begin{cases} AM^{-1}V_m g_m = \theta M^{-1}V_m g_m \\ u = M^{-1}V_m g_m \end{cases} \quad (17)$$

を考える。 u が A の固有ベクトルとなる。MORGAN(m, k) 法の場合と同様にして、式 (17) の上式の両辺に $(AM^{-1}V_m)^*$ を掛けると、

$$(AM^{-1}V_m)^* AM^{-1}V_m g_m = \theta (AM^{-1}V_m)^* M^{-1}V_m g_m \quad (18)$$

となる。ただし、

$$AM^{-1}V_m = V_{m+1}\bar{H}_m$$

の関係があるので、式 (18) の左辺の行列は、MORGAN(m, k) 法と同様な手順で、 $\bar{H}_m^* \bar{H}_m$ に置き換えられる。従って、固有ベクトル u は一般固有値問題

$$\begin{cases} \bar{H}_m^* \bar{H}_m g_m = \theta (AM^{-1}V_m)^* M^{-1}V_m g_m \\ u = M^{-1}V_m g_m \end{cases} \quad (19)$$

から求めることができる。しかし、式 (19) の行列 $(AM^{-1}V_m)^* M^{-1}V_m$ を求める計算は、直接 n 次元のベクトルの内積計算を $O(m^2)$ 回行なわねばならないので、この部分の計算コストが割高になることが難点である。

```

DEFLATION( $m, k$ )

choose  $x_0$ 
 $r_0 = b - Ax_0$ 
 $\beta = \|r_0\|, \quad v_1 = r_0/\beta$ 
 $U_l = \{\}$ 
 $l = 0$ 
 $M^{-1} = I_n$ 
start:
for  $i = 1$  to  $m$  do
   $q = AM^{-1}v_i$ 
  for  $j = 1$  to  $i$  do
     $h_{j,i} = (v_j, q)$ 
  enddo
   $\bar{v}_{i+1} = q - \sum_{j=1}^i h_{j,i}v_j$ 
   $h_{i+1,i} = \|\bar{v}_{i+1}\|, \quad v_{i+1} = \bar{v}_{i+1}/h_{i+1,i}$ 
enddo
compute  $y_m = \min_y \|\beta e_1 - \bar{H}_m y_m\|$ 
 $x_m = x_0 + M^{-1}V_m y_m$ 
 $r_m = b - AM^{-1}x_m$ 
if  $\|r_m\| / \|r_0\| < \varepsilon$  then
  stop iteration
endif
 $r_0 = r_m, \quad x_0 = x_m$ 
 $\beta = \|r_0\|, \quad v_1 = r_0/\beta$ 
compute  $f$  eigenvectors  $U_f$  from (19)
 $U_l = \text{orthog}(U_f, U_l)$ 
 $l = l + f$ 
if  $l \geq k$  then
  compute  $k$  eigenvectors  $U_k$  from (21)
   $U_l = U_k, \quad l = k$ 
endif
 $T_l = U_l^* A U_l$ 
 $M^{-1} = I_n + U_l(|\lambda_n|T_l - I_l)U_l^*$ 
goto start

```

図 4 DEFLATION(m, k) 法

DEFLATION(m, k) 法では, U_l の次元が $k + f$ となったとき, さらに固有値問題

$$\begin{cases} AU_{k+f}g_{k+f} = \theta U_{k+f}g_{k+f} \\ u = U_{k+f}g_{k+f} \end{cases} \quad (20)$$

を解く. u が A の固有ベクトルになるが, この固有値問題では, 式 (19) の一般固有値問題から求めた $k + f$ 本の固有ベクトルによって, さらに新しい固有ベクトルを求めている. これによって, 固有ベクトルの精度を改善していくことができる. MORGAN(m, k) 法の場合と同様にして, 式 (20) の上式の両辺に $(AU_{k+f})^*$ を掛けると

$$\begin{cases} (AU_{k+f})^* AU_{k+f}g_{k+f} = \theta (AU_{k+f})^* U_{k+f}g_{k+f} \\ u = U_{k+f}g_{k+f} \end{cases} \quad (21)$$

となる. 従って, 式 (21) の一般固有値問題から固有ベクトルを求めることができる. 固有値問題 (21) の小さいほうから k 個の固有値に対応する固有ベクトルが新しい固有ベクトル空間 U_k を

生成する. 固有値問題 (21) の右辺の行列 $(AU_{k+f})^*U_{k+f}$ は, $T_{k+f} = U_{k+f}^*AU_{k+f}$ の関係から, T_{k+f} に置き換えることができる. しかし, 左辺の行列 $(AU_{k+f})^*AU_{k+f}$ を求めるのに, n 次元のベクトルの内積計算を $O((k+f)^2)$ 回行なう必要がある. 図 4 に DEFLATION(m, k) 法の算法を示す. 従来の GMRES(m) 法との違いは, DEFLATED-GMRES(m, k) 法の場合とほぼ同じである. しかし, この算法では, DEFLATED-GMRES(m, k) 法と違い毎回のリスタートごとに固有値問題を 1 回ないしは 2 回解かねばならず, しかも一般固有値問題 (19) の右辺の行列と一般固有値問題 (21) の左辺の行列を求めるのに, n 次元のベクトルの内積計算をそれぞれ $O(m^2)$, $O((k+f)^2)$ 回行なう必要がある. これらの部分の計算コストが割高となる.

4 数値実験

数値実験はすべて富士通分散メモリ型並列計算機 AP3000 [12] (セル UltraSPARC 300MHz 16 台) の環境で行なった. プログラムは C 言語で記述し, 通信ライブラリーには MPI [8] を用いた. また, 固有値及び固有ベクトルを求めるルーチンには CLAPACK [6] を使用した.

4.1 各算法の性能

連立 1 次方程式を従来の GMRES(m) 法及び第 3 節で示した固有ベクトルを利用する 3 つの算法で解く実験を行なった. 収束判定条件は, $\|r_m\| / \|r_0\| < 1.0 \times 10^{-12}$ とした. 実験結果には, 収束判定条件を満たすまでに要した計算時間と反復回数を示した. また, いくつかの例について残差ノルムの収束履歴も示した.

[数値例 1] 領域 $\Omega = [0, 1] \times [0, 1] \times [0, 1]$ における次のような楕円型偏微分方程式の境界値問題を考える [9].

$$\begin{aligned} -u_{xx} - u_{yy} - u_{zz} + Ru_x &= g(x, y, z) \\ u(x, y)|_{\partial\Omega} &= 0.0 \end{aligned}$$

右辺 $g(x, y, z)$ は厳密解が

$$u(x, y, z) = \exp(xyz) \sin(\pi x) \sin(\pi y) \sin(\pi z)$$

となるように設定した. メッシュ数 $80 \times 80 \times 80$ で 7 点中心差分によって離散化した. 得られた連立 1 次方程式の次元は 512000 である. リスタート周期は 50 と 70 の 2 通りの場合を設定し, MORGAN(m, k) 法に関しては, $m+k$ が 50 もしくは 70 となるように設定した.

最初に, DEFLATED-GMRES(50,4) 法と DEFLATION(50,4) 法の場合に, 各リスタート周期で付加する固有ベクトルの数 f の値を 1 から 4 まで変化させて, 反復回数との関係を図 5 に示した. 図 5 より, もっとも反復回数の少なくなったのは DEFLATED-GMRES(50,4) 法の $R = 1000.0$ の場合を除いて, いずれの場合も $f = 1$ のときであった. この事実をもとにして, 数値例 1 では f を 1 と設定した. 次に, 実験結果を表 1 に示した. また, 残差ノルムの収束履歴を図 6 に示した.

MORGAN(m, k) 法は, R が 1.0 のときもっとも性能が良く, GMRES($m+k$) 法に比べて, 反復回数も計算時間も最大で約 3 割減少した. R が 10.0 のときでも, m が 70 の場合に計算時間は 1 割程度短縮された. しかし, R が 100.0 と 1000.0 のとき, MORGAN(m, k) 法の反復回数及び計算時間は, GMRES($m+k$) 法のそれと比べて, 2 割から 3 割増大してしまった. なお, 付加する固有ベクトルの数 k は, 4 よりも 2 のほうが効果が高かった.

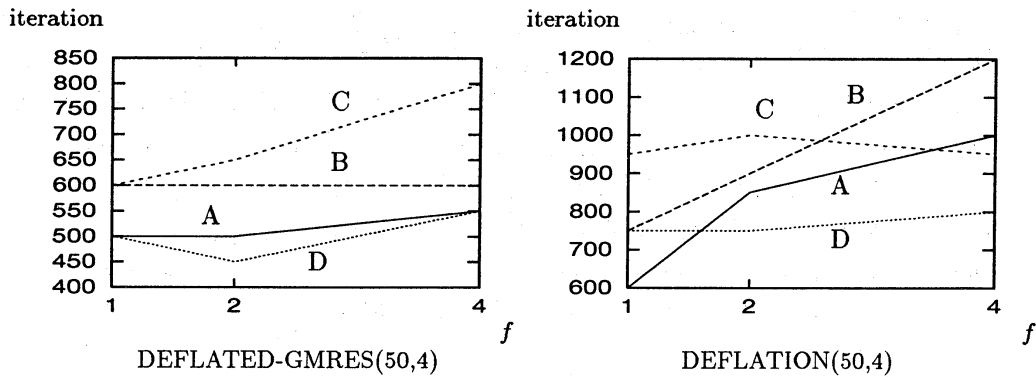


図5 数値例1のDEFLATED-GMRES(50,4)法及びDEFLATION(50,4)法における反復回数と f の関係, A: $R = 1.0$ の場合, B: $R = 10.0$ の場合, C: $R = 100.0$ の場合, D: $R = 1000.0$ の場合

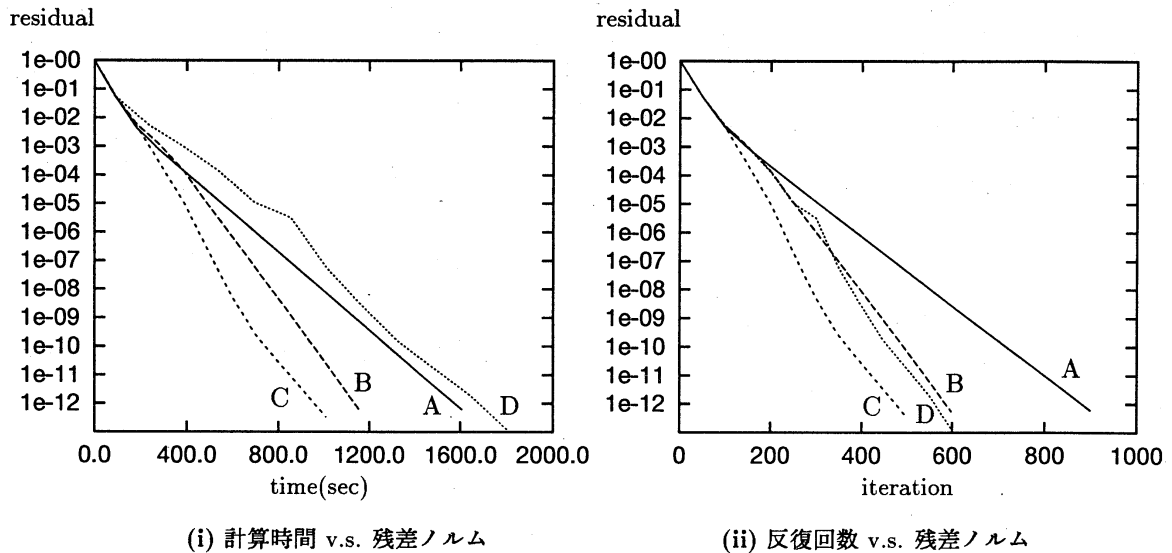


図6 数値例1に関する残差ノルムの収束履歴 ($R = 1.0$ の場合), A: GMRES(50), B: MOR-GAN(48,2), C: DEFLATED-GMRES(50,4), D: DEFLATION(50,4)

DEFLATED-GMRES(m, k)法では, k が4以上のときは, 1.0 から 1000.0 のすべての R の場合について, 同じリスタート周期の GMRES(m)法と比べて, 反復回数と計算時間が短縮された. 反復回数及び計算時間がもっとも短縮されたのは, $R = 1.0$ の m が50で k が2のときであり, いずれも4割程度短縮された. それに対して, 反復回数がまったく短縮されなかったのは, $R = 1000.0$ の m が70で k が2のときであった. しかし, このときの GMRES(m)法と比較した場合の計算時間の増大は, たかだか1割足らずであった. なお, 計算時間がもっとも短縮されたのは, 付加する固有ベクトルの数 k が6のときであった.

DEFLATION(m, k)法では, R が1.0のときに反復回数に2割から4割程度の減少がみられたが, 固有値問題を解くための計算のオーバーヘッドが高く, 計算時間の短縮には結び付かなかった. また, R が10.0以上のときは, 反復回数が1割から2割程度増大してしまい, 計算時間は最悪の場合2倍近く増大してしまった.

[数値例2] 領域 $\Omega = [0, 1] \times [0, 1] \times [0, 1]$ における次のような楕円型偏微分方程式の境界値

表 1 数値例 1 の結果 (sec: 時間 (秒) iter: 反復回数)

R	1.0		10.0		100.0		1000.0	
算法	sec	iter	sec	iter	sec	iter	sec	iter
GMRES(50)	1607.981	900	1233.826	700	1461.976	800	1183.267	650
GMRES(70)	1705.559	700	2046.644	840	2041.900	840	1694.167	700
MORGAN(48,2)	1158.015	598	1360.062	698	1845.487	948	1696.728	848
MORGAN(46,4)	1314.741	646	1419.740	696	1944.842	946	1871.948	896
MORGAN(68,2)	1462.641	558	1856.329	698	2199.449	838	2253.073	838
MORGAN(66,4)	1701.436	626	1923.713	696	2697.222	976	2452.214	906
DEFLATED-GMRES(50,2)	963.288	500	1258.420	650	1568.610	800	1171.766	600
DEFLATED-GMRES(50,4)	1008.938	500	1195.703	600	1221.820	600	1025.629	500
DEFLATED-GMRES(50,6)	1051.523	500	1266.424	600	1060.671	500	944.069	450
DEFLATED-GMRES(70,2)	1270.163	490	1425.412	560	1970.441	770	1786.639	700
DEFLATED-GMRES(70,4)	1299.760	490	1493.368	560	1490.488	560	1503.560	560
DEFLATED-GMRES(70,6)	1324.982	490	1323.513	490	1516.937	560	1318.055	490
DEFLATION(50,2)	1730.224	600	2194.283	750	2557.006	850	2631.514	900
DEFLATION(50,4)	1802.225	600	2103.354	700	2899.599	950	2258.793	750
DEFLATION(70,2)	2099.548	560	2938.354	770	3163.980	840	2905.467	770
DEFLATION(70,4)	2417.776	630	2687.695	700	3342.365	840	3568.345	910

問題を考える [4].

$$a_1 u_{xx} + a_2 u_{yy} + a_3 u_{zz} + R(a_4 u_x + a_5 u_y + a_6 u_z) + a_7 u = g(x, y, z)$$

ただし,

$$\begin{aligned} a_1 &= 2 + \sin(2\pi x) \cos(2\pi y) \cos(2\pi z) \\ a_2 &= 2 + \cos(2\pi x) \sin(2\pi y) \cos(2\pi z) \\ a_3 &= 2 + \cos(2\pi x) \cos(2\pi y) \sin(2\pi z) \\ a_4 &= \sin(4\pi x), \quad a_5 = \sin(4\pi y), \quad a_6 = \sin(4\pi z) \\ a_7 &= \sin(2\pi x) \sin(2\pi y) \sin(2\pi z) \end{aligned}$$

であり, 右辺 $g(x, y, z)$ と境界条件は厳密解が

$$u(x, y, z) = \sin(2\pi x) \cos(2\pi y) \sin(2\pi z)$$

となるように設定する. メッシュ数 $64 \times 64 \times 64$ で, 7 点中心差分によって離散化した. 得られた連立 1 次方程式の次元は 262144 である. リスタート周期は 50 と 70 の 2 通りの場合を設定した. ただし, MORGAN(m, k) 法については, $m + k$ が 50 ないしは 70 となるように設定した.

最初に, 数値例 1 と同様に DEFLATED-GMRES(50,4) 法と DEFLATION(50,4) 法について, f と反復回数の関係を図 7 に示した. 図 7 から, いずれの場合も $f = 1$ のときに反復回数は最小となった. また, $R = 100.0$ のときは, $f = 1$ のときの DEFLATED-GMRES(50,4) 法を除いていずれの場合も 70 分以内に残差ノルムは収束しなかった. 従って, この事実を元にして数値例 2 でも f を 1 と設定した. 次に, 実験結果を表 2 に示した. また, 残差ノルムの収束履歴を図 8 に示した.

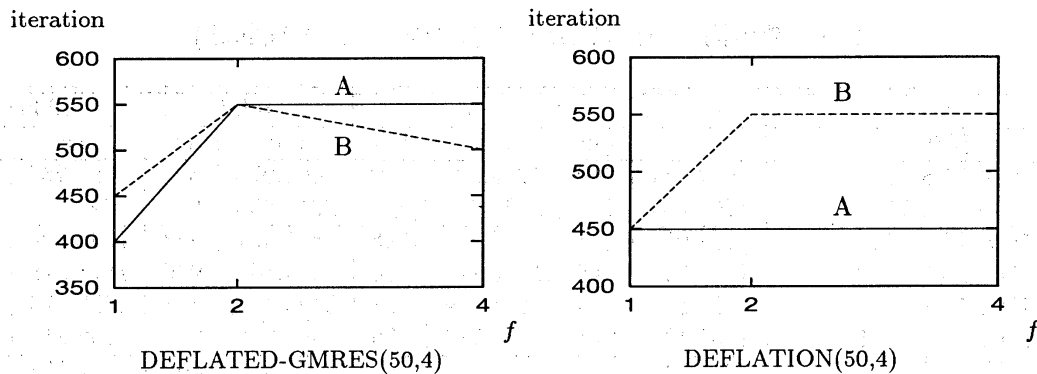


図 7 数値例 2 の DEFLATED-GMRES(50,4) 法及び DEFLATION(50,4) 法における反復回数と f の関係, A: $R = 1.0$ の場合, B: $R = 10.0$ の場合

MORGAN(m, k) 法では, k が 4 のときより 2 のときのほうが良い結果を出している. R が 1.0 と 10.0 のとき, $k = 4$ だと従来の GMRES($m+k$) 法に比べて反復回数の減少は 1 割程度みられたものの, 固有ベクトルの計算のオーバーヘッドが原因で計算時間の短縮には結びつかなかった. それに対して $k = 2$ だと, 従来の GMRES($m+k$) 法に比べて反復回数は 2 割から 3 割程度減少し, 計算時間も 1 割程度短縮された. また, R が 100.0 のとき, GMRES(50) 法及び GMRES(70) 法では, 70 分計算させても残差ノルムは収束条件を満たすことができなかった. しかし, MORGAN(m, k) 法だと, $m = 66$, $k = 4$ の場合を除いたいずれの場合でも 10 分から 11 分程度で残差ノルムは収束条件を満たした. また, この場合でも $k = 2$ のほうが $k = 4$ よりも 1 割程度計算時間が少なくなっている.

DEFLATED-GMRES(m, k) 法は, k が 4 のときに一番良い結果を出した. R が 1.0 と 10.0 のときに, GMRES(m) 法と比較して反復回数は 2 割から 3 割程度減少し, 計算時間も最大で 2 割程度短縮された. また, R が 100.0 のとき, m が 50 の場合は約 14 分, m が 70 の場合は約 8 分で残差ノルムは収束した. それに対して, k が 2 のときは, $R = 1.0, 10.0$ の場合で, GMRES(m) 法と反復回数はまったく変わらず, むしろ計算時間が 0.5 割から 1 割程度増大してしまった. また, $R = 100.0$ の場合には, 70 分の計算時間で残差ノルムが収束条件を満たさない状況もみられた. 一方, k が 6 のときは, いずれの場合も $k = 4$ のときと反復回数は変わらず, $k = 4$ の場合に対する利点はなににもなかった.

DEFLATION(m, k) 法では, R が 1.0 と 10.0 の場合に反復回数が 2 割から 3 割程度減少したものの, 固有値計算にかかるオーバーヘッドが原因で, 計算時間はむしろ GMRES(m) 法に対して 1.5 倍から 2 倍程度になってしまった. また, $R = 100.0$ のときは, DEFLATION(70,2) 法を除いて, 残差ノルムは 70 分以内に収束条件を満たさなかった.

なお, R が 1000.0 のときは, いずれの算法でも残差ノルムが収束条件を満たすことはなかった.

4.2 GMRES(m) 法に対する計算オーバーヘッド

固有ベクトルの情報を利用する 3 つの算法は, 大きく 2 つの過程に分けることができる. 1 つは GMRES(m) 法と同じ計算をする部分であり, もうひとつは固有値問題から固有ベクトルを求めたり前処理行列を算法に組み込む部分である. 本節では, 前者と後者にかかる計算時間の和を 1 周期の計算時間と呼び, 後者のことを GMRES(m) 法に対する計算オーバーヘッドと呼ぶことにする. 計算オーバーヘッドの計測の方法を各 3 つの算法について示す. MORGAN(m, k) 法は前処理

表 2 数値例 2 の結果 (sec: 時間 (秒) iter: 反復回数)

R	1.0		10.0		100.0		1000.0	
算法	sec	iter	sec	iter	sec	iter	sec	iter
GMRES(50)	523.574	550	528.664	550
GMRES(70)	620.450	490	622.656	490
MORGAN(48,2)	467.937	448	471.254	448	633.495	598
MORGAN(46,4)	541.545	496	542.083	496	705.590	646
MORGAN(68,2)	587.688	418	582.541	418	716.293	488
MORGAN(66,4)	599.677	416	707.112	486
DEFLATED-GMRES(50,2)	568.472	550	577.716	550
DEFLATED-GMRES(50,4)	428.877	400	489.673	450	820.681	750
DEFLATED-GMRES(50,6)	435.622	400	493.200	450	833.195	750
DEFLATED-GMRES(70,2)	662.998	490	669.411	490	1502.224	1120
DEFLATED-GMRES(70,4)	583.389	420	585.638	420	488.602	350
DEFLATED-GMRES(70,6)	595.581	420	594.912	420	492.266	350
DEFLATION(50,2)	764.671	500	677.408	450
DEFLATION(50,4)	709.203	450	707.520	450
DEFLATION(70,2)	848.633	420	992.573	490	3872.613	1820
DEFLATION(70,4)	844.372	420	1003.155	490

(...): 70 分の計算時間で収束条件を満たさなかった場合

を算法に組み込まないので、前処理構成にかかるオーバーヘッドはない。また、リスタート周期が $m+k$ である GMRES($m+k$) 法と比較しているの、 k 本の固有ベクトルにアーノルディ過程を適用するオーバーヘッドも考慮する必要はない。従って、MORGAN(m, k) 法の実際のオーバーヘッドは、固有値問題 (9) を解く計算のみである。DEFLATED-GMRES(m, k) 法では、固有値問題 (16) を解く計算と前処理を組み込む計算のオーバーヘッドがある。DEFLATION(m, k) 法では、二つの固有値問題 (19) と (21) を解く計算と前処理を組み込む計算のオーバーヘッドがある。3つの算法について、1 回のリスタートの計算時間に占めるこれらの計算時間の割合を表 3, 4 に示す。ただし、固有値問題を解く計算時間には、固有値問題で扱う行列を作成する計算時間も含まれるものとする。また、固有ベクトルの計算時間と前処理構成の計算時間の和を 1 周期の計算時間で割った値が、オーバーヘッドの割合である。3つの算法の中では、前処理を組み込む計算のない MORGAN(m, k) 法のオーバーヘッドがもっとも少なかった。逆に、DEFLATION(m, k) 法のオーバーヘッドはもっとも高く、いずれも 4 割程度であった。この理由は、1 回のリスタート周期で固有値問題を 2 回解かねばならないことと、固有値問題にでてくる行列を求める計算を n 次元のベクトルの内積計算によって、直接行なわねばならないためだと考えられる。これら 3つの算法では、付加する固有ベクトルの本数に比例してオーバーヘッドが増大する。それに対して、元の問題の次元が大きくなるにつれてオーバーヘッドは減少する傾向がある。従って、この 3つの算法は次元の大きい問題ほど有効であると言える。

5 終りに

本稿では固有ベクトルの情報を利用する 3つの GMRES(m) 法について考察し、AP3000 上での数値実験からそれらの算法の性能を比較評価した。

MORGAN(m, k) 法は、3つの算法の中で GMRES(m) 法に対する計算のオーバーヘッドがもっと

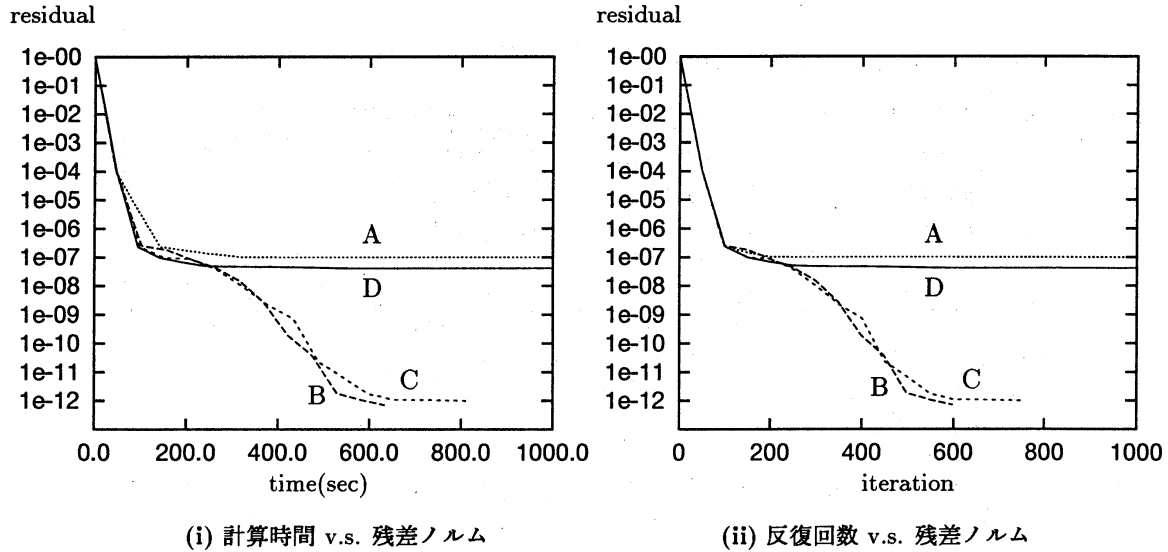


図 8 数値例 2 に関する残差ノルムの収束履歴 ($R = 100.0$ の場合), A: GMRES(50), B: MORGAN(48,2), C: DEFLATED-GMRES(50,4), D: DEFLATION(50,4)

も少ない算法であった。数値例 2 では R が 1.0 と 10.0 のときに、GMRES($m+k$) 法に比べて計算時間は 0.5 割から 1 割程度短縮され、さらに R が 100.0 のときは、GMRES($m+k$) 法では残差ノルムが 70 分で収束しなかったのに対し、MORGAN(m, k) 法では 11 分程度で収束した。しかし、数値例 1 のように、問題の係数 R が増大すると反復回数が減少してくれないケースも目立った。従って、係数 R が大きくなると、反復回数と計算時間が、GMRES($m+k$) よりもかえって増大する可能性がある。また、付加する固有ベクトルの数 k は、4 のときよりも少ない 2 のときのほうが性能は良かった。これは、アーノルディ法から固有値及び固有ベクトルを求める場合には、外側の固有値ほど元の行列の固有値を良い精度で近似することができるためである。つまり、最小固有値 (もしくは最大固有値) に近い固有値ほど良い精度で近似されるのである [2]。 $k=2$ のときは、外側から 2 つの固有値に対応する固有ベクトルのみを付加していたのに対して、 $k=4$ のときは外側から 4 つの固有値に対応する固有ベクトルを付加するので、全体として付加する固有ベクトルの精度が悪くなってしまう可能性がある。従って、今回の数値実験では、MORGAN(m, k) 法の各リスタートで付加する固有ベクトルの本数 k は 2 が最適であると言えることができる。

DEFLATED-GMRES(m, k) 法では、 k が 4 以上のときに、数値例 1 と数値例 2 のどちらの場合でも、同じリスタートの GMRES(m) 法と比べてすべての場合で反復回数と計算時間が減少した。とくに計算時間は、悪いときでも 1 割は減少し、問題によっては 4 割近く減少した。さらに、数値例 2 の係数 R が 100.0 のときのように、従来の GMRES(m) 法では 70 分間で残差ノルムが収束しなかった場合でも、8 分から 11 分の短時間で残差ノルムが収束することが確認できた。また、 k が 2 のときでも反復回数は GMRES(m) 法と最悪でも同じであり、決して増大することにはなかった。計算時間は、前処理行列を構成するためのオーバーヘッドが原因で、GMRES(m) 法よりも増大することはあったが、たかだか 1 割足らずの増大であり実用上問題のない範囲であった。また、各リスタート周期で付加する固有ベクトルの数については、 $f=1$ として最小固有値に対応する固有ベクトルのみを付加するときに、反復回数が最小になる場合が多かった。これも MORGAN(m, k) 法の場合と同様の理由で、一番外側にある最小固有値に対応する固有ベクトルのみを付加するのが、もっとも算法の性能が向上するためであると考えられる。このことは次の

表 3 数値例 1 における各リスタート周期ごとの計算オーバーヘッド (sec)

算法	固有ベクトルの計算時間	前処理構成の計算時間	1 周期の計算時間	オーバーヘッドの割合
MORGAN(48,2)	4.957	0.000	102.159	4.85 %
MORGAN(46,4)	8.696	0.000	104.690	8.31 %
MORGAN(68,2)	9.512	0.000	186.473	5.10 %
MORGAN(66,4)	15.337	0.000	193.469	7.97 %
DEFLATED-GMRES(50,2)	3.001	6.737	97.923	9.94 %
DEFLATED-GMRES(50,4)	2.888	11.583	103.239	14.02 %
DEFLATED-GMRES(70,2)	5.456	9.185	178.006	8.23 %
DEFLATED-GMRES(70,4)	5.257	16.762	190.777	11.54 %
DEFLATION(50,2)	49.576	9.429	149.048	39.59 %
DEFLATION(50,4)	50.956	14.933	153.940	42.80 %
DEFLATION(70,2)	95.880	13.559	279.981	39.09 %
DEFLATION(70,4)	97.229	21.271	284.954	41.59 %

表 4 数値例 2 における各リスタート周期ごとの計算オーバーヘッド (sec)

算法	固有ベクトルの計算時間	前処理構成の計算時間	1 周期の計算時間	オーバーヘッドの割合
MORGAN(48,2)	3.234	0.000	53.945	5.99 %
MORGAN(46,4)	5.062	0.000	54.929	9.22 %
MORGAN(68,2)	6.233	0.000	96.050	6.49 %
MORGAN(66,4)	9.911	0.000	106.487	9.31 %
DEFLATED-GMRES(50,2)	1.798	3.533	51.563	10.34 %
DEFLATED-GMRES(50,4)	1.983	6.562	55.395	15.43 %
DEFLATED-GMRES(70,2)	3.485	4.871	96.219	8.68 %
DEFLATED-GMRES(70,4)	3.691	8.713	103.598	11.97 %
DEFLATION(50,2)	28.065	4.819	79.582	41.32 %
DEFLATION(50,4)	28.622	7.576	81.698	44.31 %
DEFLATION(70,2)	54.303	6.783	150.846	40.50 %
DEFLATION(70,4)	55.979	10.632	156.773	42.49 %

DEFLATION(m, k) 法についてもあてはまる。

DEFLATION(m, k) 法では、反復回数に減少がある場合でも、GMRES(m) 法に対する計算のオーバーヘッドが 40 % もあるので、計算時間の短縮には結び付かなかった。オーバーヘッドを考慮すると、GMRES(m) 法に比べて反復回数の減少は最低でも 50 % は必要だと思われる。また、MORGAN(m, k) 法と同様、数値例 1 で R が大きいときに反復回数の増大がみられた。従って、固有値問題 (21) を解くことで必ずしも固有ベクトルの精度を改善することができるとは限らないといえる。むしろ、逆に精度に悪影響を及ぼすこともあると考えられる。

以上を総合すると、数値例 1 及び数値例 2 のすべての場合において、同じリスタート周期を持つ GMRES(m) 法と比較して、計算時間と反復回数のどちらも減少した DEFLATED-GMRES(m, k) 法 (ただし、 $k \geq 4$ のとき) が、3つの算法の中でもっとも有効である。また、 $f = 1$ として、各リスタートで付加する固有ベクトルを最小固有値に対応するものだけにする場合に、DEFLATED-GMRES(m, k) 法は多くの場合でもっとも性能を発揮できると言える。しかし、DEFLATED-

GMRES(m, k) 法では、アーノルディ過程によって固有値問題を解いているので、精度の良い固有ベクトルを求めるのにリスタート周期を長くとることが要求され [2], 直交化にかかる計算時間が増大することが難点である。従って、このような難点を改善するような手法を考えることが今後の課題として挙げられる。

参考文献

- [1] Hestens, M. R. and Stiefel, E.: Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards*, Vol. 49, pp. 409–435 (1952).
- [2] J. H. Wilkinson, *The algebraic eigenvalue problem*, Clarendon Press, Oxford, (1965).
- [3] Y. Saad and M. K. Schultz.: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, No. 7, pp. 856–869, (1986).
- [4] Schönauer, W.: Scientific Computing on Vector Computers, North Holland (1987).
- [5] R. Morgan.: Computing Interior Eigenvalues of Eigenvalues of Large Matrices, *Numer. Linear Algebra Appl.*, No. 154–156, pp. 289–309, (1991).
- [6] 小国 力 訳.: 『行列演算パッケージ LAPACK 利用の手引』, 丸善, (1993).
(<http://www.netlib.org/{lapack,clapack}/index.html>)
- [7] R. Morgan.: A restarted GMRES method augmented with eigenvectors, *SIAM J. Matrix Anal. App.*, No. 16, pp. 1154–1171, (1995).
- [8] Message Passing Interface Forum, MPI: A Message-Passing Standard, May (1995).
(<http://www.mcs.anl.gov/mpi/>)
- [9] Sleijpen, G. L. G. and Van der Vorst, H. A.: An overview of approaches for the stable computation of hybrid BiCG methods, *Appl. Numer. Math.*, Vol. 19, pp. 235–254 (1995).
- [10] J. Erhel, K. Burrage and B. Pohl.: Restarted GMRES preconditioned by deflation, *Journal of Computational and Applied Mathematics*, No. 69, pp. 303–318, (1996).
- [11] K. Burrage and J. Erhel.: On the Performance of Various Adaptive Preconditioned GMRES Strategies, *Numer. Linear Algebra Appl.*, No. 5, pp. 101–121, (1998).
- [12] 富士通研究所 : AP3000: Products: Hardware, http://www.fujitsu.co.jp/hypertext/Products/Info_process/hpc/ap3000/products/hw.html .